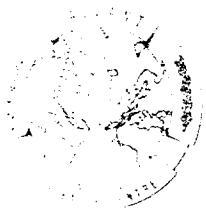
SACLANTCEN MEMORANDUM
serial no.: SM-207

SACLANT UNDERSEA
RESEARCH CENTRE

MEMORANDUM



MAKE_MAP and MEDMAP:

Two programs for plotting maps of the Mediterranean Sea

P. Scrimger and
A. Trangeled

June 1988

The SACLANT Undersea Research Centre provides the Supreme Allied Commander Atlantic (SACLANT) with scientific and technical assistance under the terms of its NATO charter, which entered into force on 1 February 1963. Without prejudice to this main task—and under the policy direction of SACLANT—the Centre also renders scientific and technical assistance to the individual NATO nations.

Page count for SM-207
(excluding covers)

Pages	Total
i–iv	4
1–20	20
	24

SACLANT Undersea Research Centre
Viale San Bartolomeo 400
19026 San Bartolomeo (SP), Italy

tel: 0187 540 111
telex: 271148 SACENT I

NORTH ATLANTIC TREATY ORGANIZATION

SACLANTCEN SM-207

MAKE_MAP and MEDMAP:

Two programs for plotting maps of the Mediterranean Sea

P. Scrimger and A. Trangeled

The content of this document pertains to work performed under Project 21 of the SACLANTCEN Programme of Work. The document has been approved for release by The Director, SACLANTCEN.

Issued by:
Underwater Research Division

R. Thiele
Division Chief

intentionally blank page

MAKE_MAP and MEDMAP:

Two programs for plotting maps of the Mediterranean Sea

P. Scrimger and A. Trangeled

Abstract: Two FORTRAN programs MAKE_MAP and MEDMAP are described which, when used together, will plot maps of all or any portion of the Mediterranean Sea. Examples are given which show the high degree of detail provided by the 2′ resolution of the database. A description of how the maps are created in the MAKE_MAP program by means of intermediate landmass matrices is given, and applications of these landmass matrices are mentioned. A flow chart of the main stages of this program is given The landmass matrix is read by MEDMAP which uses an interpolating contour routine to plot the coastline; a flow chart of the program is given. FORTRAN listings for MAKE_MAP and MEDMAP are also included.

Keywords: maps o Mediterranean o modelling o SONDA o UNIRAS

Contents

1. Introduction

There is an ever-present need in any major research organization to be able to quickly and easily display graphical information. At SACLANTCEN the SONDA [1,2] system was used in past years to display such oceanographic information. This report presents two new computer programs which provide similar plotting capabilities to that provided by the SONDA system for areas located in the Mediterranean Sea. The advantages offered by these new programs are the increased resolution and the creation of intermediate plotting matrices which can be used in modelling applications. The MAKE_MAP and MEDMAP programs were originally developed to satisfy the requirements of an existing model in use at SACLANTCEN, namely the Mediterranean shipping distribution model [3]; however they can easily find application in other areas.

Section 2 gives three examples of map production using the available database, these examples include plots of the entire Mediterranean Sea, the central Mediterranean including the islands of Corsica, Sardinia and Sicily as well as the Italian coastline and finally the Aegean Sea. Section 3 describes the MAKE_MAP program and provides information on how the pre-plotting data matrix is created and Sect. 4 describes the MEDMAP program and lists the various output devices supported by the software. The two programs have been written in VAX FORTRAN and are currently running on a VAX 8600 with the VAX/VMS version 4.6 operating system. They are listed in Appendix A and Appendix B, respectively. The library plotting routines are all taken from the commercially-available graphics package UNIRAS.

2. Some sample maps

The examples shown in this section are designed to show the usefulness and ease of operation of the two programs. The first example produces a map of the whole Mediterranean Sea. It makes use of the entire mapping database defined from (30°N,6°W) to (46°N,37°E) in 2′ steps. This database was originally created from a series of 10 charts of the Mediterranean [4] drawn using a Mercator projection at a scale of 1:1 000 000 at 38°N.

In order to generate one of these maps, the user should follow these steps in VAX/VMS DCL:

```
1.  $ DEFINE DATA Device_1:[Directory_1]
2.  $ RUN Device_2:[Directory_2]MAKE_MAP
3.  $ RUN Device_2:[Directory_2]MEDMAP
```

Note that in step 1 the user must define the logical name 'DATA' to point to the directory which is to contain the intermediate data matrix created by the program MAKE_MAP. Step 2 will run the MAKE_MAP program which creates the intermediate data matrix (see Sect. 3) and step 3 will run the MEDMAP program which plots this data matrix (see Sect. 4).

Example 1

Figure 1 was generated by entering the following data in response to the prompts issued by program MAKE_MAP:

```
Enter coordinates of lower left cell (min.  30N06W):   30N06W
Enter coordinates of upper right cell (max.  46N37E):  46N37E
```

This causes the MAKE_MAP program to read the entire database which consists of 688 separate input files. Each input file corresponds to a 1° × 1° area of the Mediterranean Sea and contains 900 elements (−1.0 or 0.0) contained in 30 records of 30 elements per record. The plotting is done by program MEDMAP on the user selected output device.

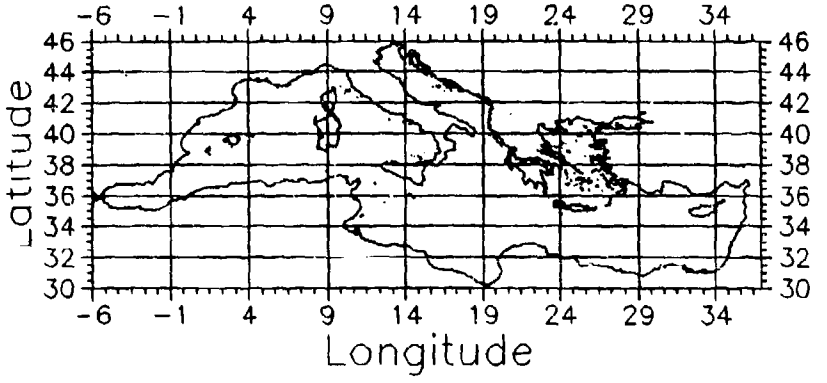


Fig. 1. The Mediterranean Sea.

Example 2

Example 2 reads a subsection of the database which contains the islands of Corsica, Sardinia and Sicily as well as the Italian coastline. The water masses include the Tyrrhenian Sea, the Ligurian Sea and the Adriatic Sea as well as the northern portion of the Ionian Sea. This area is defined from (36°N,7°E) to (46°N,20°E) and so the following data was entered in response to the prompts issued by program MAKE_MAP:

```
Enter coordinates of lower left cell (min.  30N06W):   36N07E
Enter coordinates of upper right cell (max.  46N37E):  46N20E
```

Note that in this example only 130 out of the total 688 input files are used when generating an intermediate plotting matrix. The resulting plot is shown in Fig. 2.

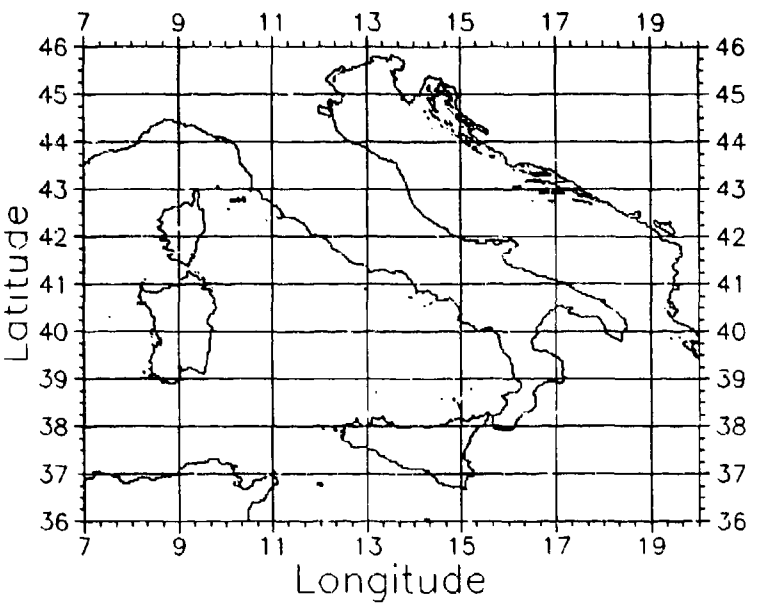


Fig. 2. **The Central Mediterranean.**

Example 3

Example 3 again uses a subsection of the database and illustrates some of the fine detail available at a resolution of 2 min. The map area is defined from (35°N,22°E) to (42°N,29°E), and so the following data was entered in response to the prompts issued by program MAKE_MAP:

```
Enter coordinates of lower left cell (min.  30N06W):   35N22E
Enter coordinates of upper right cell (max.  46N37E):  42N29E
```

Note that in this example 49 out of the total 688 input files are used when generating an intermediate plotting matrix. The resulting plot is shown in Fig. 3.

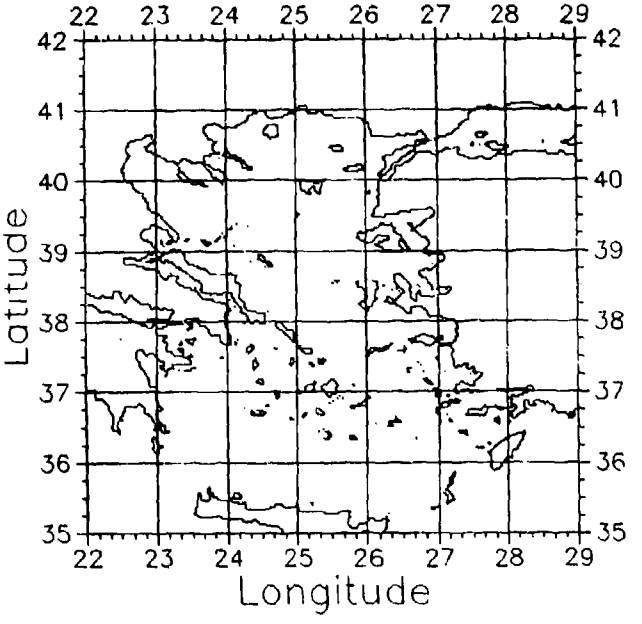


Fig. 3. The Aegean Sea.

3. The MAKE_MAP program

The function of the MAKE_MAP program is to create a data matrix in a form which is suitable for subsequent plotting with the MEDMAP program. There are two important advantages to using such a two-stage process in obtaining plots. The first advantage is plotting speed since each subsequent plot (via MEDMAP) can be done without having to reaccess the full database. This is useful for example when the user wishes to preview the plot on his terminal before obtaining a hard copy (on a colour plotter, laser printer, etc.). A second advantage is that an intermediate landmass matrix is produced. This matrix is made up of $(2' \times 2')$ cells containing either a -1.0 (land) or a 0.0 (water). Landmass matrices of this type are often used in modelling applications, for example when modelling shipping movements [1], or modelling target locations [5] and could find future application in modifications to such programs as the RANDI-2 ambient noise model [6] where the position of landmasses could be used to identify the end points of the transmission loss function. The organizational layout of this program is shown schematically in the flow chart given in Fig. 4.

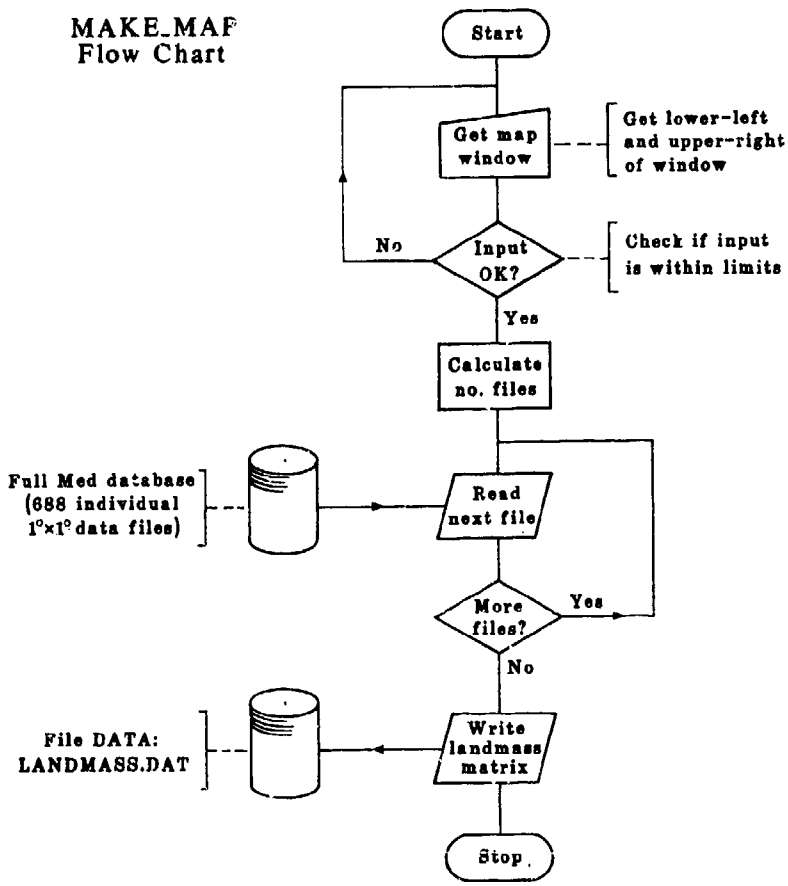


Fig. 4. A flow chart of program MAKE_MAP.

4. The MEDMAP program

This program represents the second stage of map production and is primarily a plotting routine designed to allow the user to produce output on the device of his choosing. The program uses a 2D contour mapping routine which supports 10 levels of interpolative smoothing. The presently supported devices at SACLANTCEN are presented to the user as a 'form'. A copy of this form is given in Fig. 5.

MEDMAP V1.0			Seldev V2.0
Hardcopy devices		CRT devices	
COL	Tektronix 4691 A3	VTT	Local Vt200-series
COL4	Tektronix 4691 A4	LTEK	Local Tek41XX-series
VUG	Tektronix 4692 Vugraf	4105	Host Tek4105
PRX	Printronix OPER-room		
PRXU	Printronix user area	NEWS	DSI news of 01-NOV-1987
CCP	Calcomp 5105	INFO	UNIRAS info
LA50	LA50 Printer	DELA	Delete completed plots
T03	LN03 to USR$LASER	UTIL	Soon available
		EXIT	Terminates image
Select output device:		(Press PF2 for HELP)	

Fig. 5. The plotting options form.

As before, the organizational layout of the program is shown schematically in a flow chart (Fig. 6).

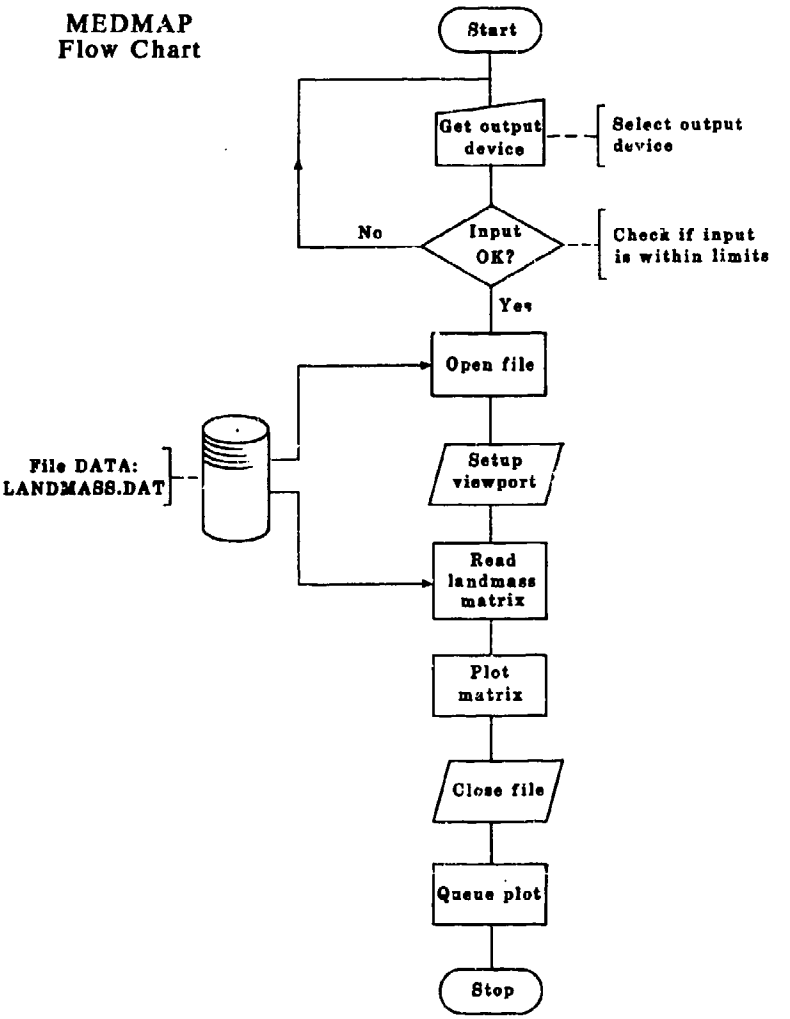


Fig. 6. A flow chart of program MEDMAP.

References

[1] WINTERBURN, R.F.J. The SACLANTCEN oceanographic database, Volume I: Design criteria and data structure and content, SACLANTCEN SM-150. La Spezia, Italy, SACLANT ASW Research Centre, 1981.

[2] WINTERBURN, R.F.J. The SACLANTCEN oceanographic database, Volume II: Access, interrogation and display, SACLANTCEN SM-151. La Spezia, Italy, SACLANT ASW Research Centre, 1981.

[3] SCRIMGER, P. and HEITMEYER, R.M. A computer model of the movement of shipping in a basin with application to the Mediterranean Sea, SACLANTCEN SR-143, La Spezia, Italy, SACLANT Undersea Research Centre, 1988.

[4] INTERGOVERNMENTAL OCEANOGRAPHIC COMMISSION, International bathymetric chart of the Mediterranean, Leningrad, USSR, Department of Navigation and Oceanography, Ministry of Defence (Under the authority of IOC (UNESCO)), 1981.

[5] HEITMEYER, C. and CARRIERE, J.-C. A hybrid approach to multitarget tracking. SACLANTCEN SR-130, La Spezia, Italy, SACLANT Undersea Research Centre, 1988.

[6] HAMSON, R.M. and WAGSTAFF, R.A. An ambient-noise model that includes coherent hydrophone summation for sonar system performance in shallow water, SACLANTCEN SR-70. La Spezia, Italy, SACLANT ASW Research Centre, 1983.

Appendix A

MAKE_MAP, a FORTRAN listing

```
1          PROGRAM MAKE_MAP
2
3  C       PROGRAM DESCRIPTION:
4  C
5  C               Creates file DATA:LANDMASS.DAT from data taken from
6  C           US6:[SCRINGER.WORK]. The output file will contain coastline
7  C           data for a selected area.
8  C
9  C       AUTHORS:
10 C
11 C
12 C               Alex Trangeled & Paul Scrimger
13 C               SACLANT Undersea Research Center,
14 C               V. San Bartolomeo 400,
15 C               19026 La Spezia, Italy
16 C
17 C       CREATION DATE:   Summer 1987
18 C
19 C
20 C
21 C                       C H A N G E   L O G
22 C
23 C       Date      | Name  | Description
24 C---------------+-------+-----------------------------------------------------
25 C[change_entry]
26 C
27
28          COMMON /POS/ LAT1,LAT2,LNG1,LNG2
29
30          CHARACTER*1      CSTR,NSTR,ESTR,WSTR
31          CHARACTER*2      STRLAT,STRLNG
32          CHARACTER*10     PROMPT
33          CHARACTER*40     OUTPUT_FILE,BUFF1,BUFF2,LINE,FIL
34
35          INTEGER*4 PARSE_POS,ARRAY(30,30)
36
37          REAL             MATRIX(1290,480)
38
39 C        ------------------------------------------------------------
40          PROMPT = '($,1X,A)'
41          CSTR = 'C'
42          OUTPUT_FILE = 'DATA:landmass'
43
44 C        ------------------------------------------------------------
45          OPEN (UNIT=9,FILE=OUTPUT_FILE,STATUS='NEW',
46          1FORM='UNFORMATTED',ERR=999)
47
48 10       WRITE(6,PROMPT) 'Enter coordinates of lower left cell
49          1(min. 30N06W): '
50
```

```
      READ(5,'(A)',END=10) BUFF1
      ISTAT=PARSE_POS(BUFF1,1)
      IF(ISTAT.NE.0) GOTO 10

C     ------------------------------------------------------------
 20   WRITE(6,PROMPT) 'Enter coordinates of upper right cell
     1(max. 46N37E): '

      READ(5,'(A)',END=20) BUFF2
      ISTAT=PARSE_POS(BUFF2,2)
      IF(ISTAT.NE.0) GOTO 20

C     ------------------------------------------------------------
C     Calculate no. of cells in x direction (longitude)


      NO_CELLX=(LNG2-LNG1)+1
      NO_ELEMX=NO_CELLX*30     !30*30 data points in each cell

C     Calculate no. of cells in y direction (latitude)

      NO_CELLY=(LAT2-LAT1)+1
      NO_ELEMY=NO_CELLY*30


C     Set MATRIX counter to 0 in x and y direction

      INATX=0
      INATY=0

C     Start reading data and fill MATRIX

      ITOTAL_FILES=(LAT2-LAT1+1)*(LNG2-LNG1+1)

C     ------------------------------------------------------------
      DO 110 LAT=LAT1,LAT2
          DO 100 LNG=LNG1,LNG2

              ICUR_FIL=ICUR_FIL+1

              IF(LNG.LE.0) THEN
                  WRITE(FIL,800) LAT,LNG*-1
              ELSE
                  WRITE(FIL,801) LAT,LNG
              ENDIF


C     Open cell-file and fill up array with 30*30 elements

          OPEN(UNIT=10,FILE='US6:[SCRINGER.WORK]'//FIL,READONLY,STATUS='OLD',
     1    ERR=900)

          WRITE(6,666) fil(1:10),ICUR_FIL,ITOTAL_FILES
          DO II=30,1,-1
              READ(10,'(A40)') LINE
              DO JJ=6,35
                  READ(LINE(JJ:JJ),'(I1)') ARRAY(JJ-5,II)
              END DO
          END DO
```

```
111               CLOSE(UNIT=10,STATUS='KEEP')
112
113   C       Write these elements to the big array
114
115               DO II=1,30
116                   DO jj=1,30
117                       MATRIX(JJ+IMATX,II+IMATY)=ARRAY(JJ,II)*-1.
118                   END DO
119               END DO
120               IMATX=IMATX+30
121
122   100         CONTINUE
123               IMATX=0
124               IMATY=IMATY+30
125   110     CONTINUE
126
127   C       ---------------------------------------------------------------
128
129   800     CONTINUE
130
131   C       Write range information to output file
132
133           WRITE(9) NO_ELEMX
134           WRITE(9) NO_ELEMY
135           WRITE(9) LNG1
136           WRITE(9) LNG2+1
137           WRITE(9) LAT1
138           WRITE(9) LAT2+1
139
140
141           WRITE (9) ((MATRIX(I,J),I=1,NO_ELEMX),J=1,NO_ELEMY)
142           CLOSE (UNIT=9,STATUS='KEEP')
143           STOP
144   C       ---------------------------------------------------------------
145
146   900     WRITE(6,'(1X,A)') 'ERROR OPENING FILE '//FIL
147           STOP
148   C       ---------------------------------------------------------------
149
150   999     STOP 'Error opening outfile'
151
152   C       ---------------------------------------------------------------
153   666     FORMAT('+Current file ',A10,'is no. ',i3,' out of ',i3,' files')
154   800     FORMAT(1X,'C',I2.2,'N',I2.2,'W')
155   801     FORMAT(1X,'C',I2.2,'N',I2.2,'E')
156
157           END
158
159           INTEGER FUNCTION PARSE_POS(LOWBUFF,ITYPE)
160   C       ROUTINE DESCRIPTION:
161   C
162   C               This routine parses the coordinate passed in LOWBUFF
163   C               If value is illegal parse_pos=1 else parse_pos=0
164   C               ITYPE specifies if we are reading first or second value
165   C               If parse_pos succeeds, the common block POS is loaded with the
166   C               appropiate values
167   C
```

```
C       AUTHORS:
C
C               Alex Trangeled & Paul Scrimger
C               SACLANT Undersea Research Center,
C               V. San Bartolomeo 400,
C               19026 La Spezia, Italy
C
C       CREATION DATE:   Summer 1987
C
C
C                       C H A N G E   L O G
C
C       Date      | Name  | Description
C---------------+-------+-----------------------------------------------------
C[change_entry]
C
        COMMON /POS/ LAT1,LAT2,LNG1,LNG2

        CHARACTER*40 BUFF,LOWBUFF
C       ------------------------------------------------------------
        PARSE_POS=0

        CALL STR$UPCASE(BUFF,LOWBUFF)   !Convert string to uppercase

        READ(BUFF(1:2),'(I2)',ERR=50) ITEMP1 !Read integer value
        READ(BUFF(4:5),'(I2)',ERR=50) ITEMP2

        IF  (BUFF(3:3).NE.'N') GOTO 50
        IF  ((BUFF(6:6).NE.'E').AND.(BUFF(6:6).NE.'W')) GOTO 50

C       ------------------------------------------------------------
        PARSE_POS=0     !Everything ok
        IF(ITYPE.EQ.1) THEN
            LAT1=ITEMP1
            LNG1=ITEMP2
            IF(BUFF(6:6).EQ.'W') LNG1=LNG1*-1

        ELSE IF(ITYPE.EQ.2) THEN

            LAT2=ITEMP1-1                           !29-MAY-1987
            LNG2=ITEMP2-1                           !29-MAY-1987
            IF(BUFF(6:6).EQ.'W') LNG2=LNG2*-1

        ELSE

            WRITE(6,'(1X,A)') 'Illegal type specified -
     1      Check your program !!!'//CHAR(7)
            STOP 'termination on error'

        END IF
        RETURN
C       ------------------------------------------------------------

50      PARSE_POS=1     !Invalid input
        WRITE(6,'(1X,A)') 'Illegal coordinate specified -
     1Please reenter !!!'//char(7)
        RETURN

        END
```

Appendix B

MEDMAP, a FORTRAN listing

```
1         PROGRAM MEDMAP
2  C
3  C      PROGRAM DESCRIPTION:
4  C
5  C           This program plots the coastline data contained in the file
6  C       DATA:LANDMASS.DAT, which is created by MAKE_MAP. For additional
7  C       information please refer to separate documentation.
8  C
9  C           Link this program using the following command:
10 C
11 C           $ LINK MEDMAP, US6:[TRANGELED.SUBS]ROUTINES/LIB,SL:UNIRAS/LIB
12 C
13 C      AUTHORS:
14 C
15 C           Alex Trangeled & Paul Scringer
16 C           SACLANT Undersea Research Center,
17 C           V. San Bartolomeo 400,
18 C           19026 La Spezia, Italy
19 C
20 C      CREATION DATE:   Summer 1987
21 C
22 C
23 C                    C H A N G E   L O G
24 C
25 C      Date      | Name  | Description
26 C---------------+-------+-------------------------------------------------
27 C[change_entry]
28 C
29
30
31        INTEGER         STATUS,         !Status returned by system calls
32        2               VM_SIZE,        !Size of virtual memory (VM) needed
33        2               VM_ADDR,        !Starting address of the VM
34        2               MSDICOL         !Declaration of the main part
35
36        INTEGER         LIB$GET_VM,     !System routines - for documentation
37        2               LIB$FREE_VM,    !see VAX/VMS System Services reference
38        2               LIB$SHOW_VM     !guide
39
40
41        CALL LIB$INIT_TIMER
42
43        CALL GROUTE('LIST *')   !Prompt for output device
44
45 C      ----------------------------------------------------------------
46 C      Open the LANDMASS file to find the number of elements in it
47
48        OPEN(UNIT=9,ERR=9999,FILE='DATA:LANDMASS.DAT',
49        1STATUS='OLD',FORM='UNFORMATTED',READONLY)
50
51        READ (9) L1     !No. elements in X
52        READ (9) L2     !No. of elements in Y
```

```
53
54          !Close the LANDMASS file
55
56          CLOSE (UNIT=9,STATUS='KEEP')
57  C       -------------------------------------------------------------
58  C       Calculate the amount of VM we need for this file. We'll
59  C       need 4 bytes for every element in the matrix.
60
61          VM_SIZE=L1*L2*4 !Size we will need
62
63  C       Allocate the VM
64
65          STATUS=LIB$GET_VM(VM_SIZE,VM_ADDR)
66          IF(.NOT.STATUS) CALL LIB$SIGNAL(%VAL(STATUS))
67
68  C       Call the map drawing part, passing the starting address
69  C       and size of the VM that we allocated
70
71          STATUS=MSD1COL(%VAL(VM_ADDR),VM_SIZE)
72          IF(.NOT.STATUS) CALL LIB$SIGNAL(%VAL(STATUS))
73
74  C       Deallocate the VM
75
76          STATUS=LIB$FREE_VM(VM_SIZE,VM_ADDR)
77          IF(.NOT.STATUS) CALL LIB$SIGNAL(%VAL(STATUS))
78
79  C       Terminate UNIRAS
80
81          CALL GCLOSE
82
83          CALL LIB$SHOW_TIMER
84          CALL EXIT(0)
85
86          STOP 'Normal successful termination'
87  9999    STOP 'Error opening landmass database'
88
89          END
90
91          INTEGER FUNCTION MSD1COL(Z,IZSIZE)
92
93          DIMENSION   Z(izsize),      !VM area containing shipping matrix
94          1           X(600),         !The X- and Y-array are used with
95          1           Y(600)          !the ship (or sub) tracks
96          DIMENSION   IZ(2)           !Class limit array
97
98          CHARACTER*12    FIL1        !File name string
99          CHARACTER*12    NSTRING
100         CHARACTER*1     YESNO
101
102         LOGICAL SEGSTORE,GRID
103
104         SEGSTORE = .FALSE.          !Set flag if UNIRAS segment storage is on
105         GRID = .FALSE.
106         XOR=20.
107         YOR=30.
108
109         !Prompt for the size in the x direction
110
```

```
111           CALL dsi$edstr (
112           1         'Enter XS in mm: ',
113           1         '200.',
114           1         NSTRING)
115
116           READ(NSTRING,*) XS
117
118
119
120
121           IF (XS.EQ.0.) THEN
122               WRITE(6,*) 'XS NOT SPECIFIED - USING DEFAULT'
123               XS=200.
124           ELSE
125               WRITE(6,*) 'XS SPECIFIED - ',XS
126           END IF
127
128           CALL dsi$edstr (
129           1         'Do you want to plot a grid Y/N: ',
130           1         'Y',
131           1         YESNO)
132
133           IF((YESNO.EQ.'Y').OR.(YESNO.EQ.'y')) GRID = .TRUE.
134
135           CALL dsi$edstr (
136           1         'Do you want to create a UNIPICT file Y/N: ',
137           1         'Y',
138           1         YESNO)
139
140           IF((YESNO.EQ.'Y').OR.(YESNO.EQ.'y')) SEGSTORE = .TRUE.
141
142
143           YS=XS*0.8
144
145           ITX=1
146           ITY=1
147
148   C       Define only one class limit as we want to use two colours only
149
150           ZZ(1)=-0.9
151
152   C       ------------------------------------------------------------
153   C       Read binary file
154
155           OPEN(UNIT=9,ERR=9999,FILE='DATA:LANDMASS.DAT',
156           1STATUS='OLD',FORM='UNFORMATTED',READONLY)
157
158           READ (9) MAXX
159           READ (9) MAXY
160           READ (9) LNG1
161           READ (9) LNG2
162           READ (9) LAT1
163           READ (9) LAT2
164
165           LNG2=LNG2
166           LAT2=LAT2
167
168           !Print info to terminal
169
```

```
170           WRITE(6,800) MAXX
171           WRITE(6,801) MAXY
172           WRITE(6,802) LNG1
173           WRITE(6,803) LNG2
174           WRITE(6,804) LAT1
175           WRITE(6,805) LAT2
176
177           READ (9) (Z(J),J=1,MAXX*MAXY)
178           CLOSE (UNIT=9,STATUS='KEEP')
179
180   C       ---------------------------------------------------------------
181   C       Initiate UNIRAS
182
183           CALL GOPEN
184           CALL GRESET
185
186           IF (SEGSTORE) THEN
187               WRITE(6,*) 'Segment storage is ACTIVE'
188               WRITE(6,*) 'Opening segment file #1! '
189               CALL GSEGCR(1)
190           END IF
191
192
193   C       Set up class limits, user coordinate system and viewport
194
195           CALL GZCL(ZZ,1,0)
196           CALL GLIMIT(FLOAT(LNG1),FLOAT(LNG2),FLOAT(LAT1),FLOAT(LAT2),0.,0.)
197           CALL GVPORT(XOR,YOR,XS,YS)
198
199   C       Set colour of contourlines to anti-background
200
201           CALL GEOCOL(1)
202
203   C       Set smoothing level and plot contour lines
204
205           CALL GSMTH(9)
206           CALL GCNR2V(Z,MAXX,MAXY)
207
208   C       Terminate GCNR2V
209   C       And draw axis
210
211           ITX=ITX*6
212           ITY=ITY*4
213
214           CALL GLIMIT(FLOAT(LNG1),FLOAT(LNG2),FLOAT(LAT1),FLOAT(LAT2),0.,0.)
215
216           CALL GTICKN(ITX)
217           CALL GAXIS(1,FLOAT(LNG1),0.,FLOAT(LNG2),'Longitude$')
218
219           CALL GTICKN(ITY)
220           CALL GAXIS(2,FLOAT(LAT1),0.,FLOAT(LAT2),'Latitude$')
221
222   C       Draw secondary axis
223
224           CALL GAXORI(FLOAT(LNG2),FLOAT(LAT1))
225           CALL GTICKN(ITY)
226           CALL GAXIS(-2,FLOAT(LAT1),0.,FLOAT(LAT2),'$')
```

```
227
228           CALL GAXORI(FLOAT(LNG1),FLOAT(LAT2))
229           CALL GTICKN(ITI)
230           CALL GAXIS(-1,FLOAT(LNG1),0.,FLOAT(LNG2),'$')
231
232           CALL GEOCOL(1)
233
234           IF (GRID) CALL GGRID(1,1)
235           CALL GUNDEF(999.999,31)
236
237           IF (SEGSTORE) CALL GSEGCL(1)
238           CALL GCHARJ(0)
239           medicol=1
240
241           RETURN
242
243
244   800     FORMAT(1X,'No. points in the X-direction: ',I)
245   801     FORMAT(1X,'No. points in the Y-direction: ',I)
246   802     FORMAT(1X,'Longitude minimum              :',I)
247   803     FORMAT(1X,'Longitude maximum              :',I)
248   804     FORMAT(1X,'Latitude minimum               :',I)
249   805     FORMAT(1X,'Latitude maximum               :',I)
250
251   9998    STOP 'Error during read of track data base'
252   9999    STOP 'Error opening landmass data base'
253
254           END
255
256           SUBROUTINE DSI$EDSTR(PROMPT,DEFAULT,ANSWER)
257   C
258   C       ROUTINE DESCRIPTION:
259   C
260   C               This routine writes PROMPT on the terminal, and
261   C               allows the user to use VMS's line editing functions to
262   C               modify or replace the default answer
263   C
264   C       AUTHORS:
265   C
266   C               Alex Trangeled
267   C               SACLANT Undersea Research Center,
268   C               V. San Bartolomeo 400,
269   C               19026 La Spezia, Italy
270   C
271   C       CREATION DATE:   May 1986
272   C
273   C
274   C                       C H A N G E   L O G
275   C
276   C       Date     | Name  | Description
277   C---------------+-------+-----------------------------------------------------
278   C[change_entry]
279   C
280           CHARACTER*(*)   PROMPT,
281           1               DEFAULT,
282           1               ANSWER
283           COMMON /SMGID/  ID
284
285           IF (ID.NE.0) GOTO 5
```

```
286
287           CALL SMG$CREATE_VIRTUAL_KEYBOARD(ID)
288  5        CALL SMG$READ_STRING(ID,ANSWER,PROMPT,
289        1                      ,,,,,,,DEFAULT)
290           RETURN
291           END
292
293
```

Initial Distribution for SM-207

Ministries of Defence	
JSPHQ Belgium	2
DND Canada	10
CHOD Denmark	8
MOD France	8
MOD Germany	15
MOD Greece	11
MOD Italy	10
MOD Netherlands	12
CHOD Norway	10
MOD Portugal	2
MOD Spain	2
MOD Turkey	5
MOD UK	20
SECDEF US	60
NATO Authorities	
Defence Planning Committee	3
NAMILCOM	2
SACLANT	3
SACLANTREPEUR	1
CINCWESTLANT/ COMOCEANLANT	1
COMSTRIKFLTANT	1
CINCIBERLANT	1
CINCEASTLANT	1
COMSUBACLANT	1
COMMAIREASTLANT	1
SACEUR	2
CINCNORTH	1
CINCSOUTH	1
COMNAVSOUTH	1
COMSTRIKFORSOUTH	1
COMEDCENT	1
COMMARAIRMED	1
CINCHAN	3
SCNR for SACLANTCEN	
SCNR Belgium	1
SCNR Canada	1
SCNR Denmark	1
SCNR Germany	1
SCNR Greece	1
SCNR Italy	1
SCNR Netherlands	1
SCNR Norway	1
SCNR Portugal	1
SCNR Turkey	1
SCNR UK	1
SCNR US	2
French Delegate	1
SECGEN Rep. SCNR	1
NAMILCOM Rep. SCNR	1
National Liaison Officers	
NLO Canada	1
NLO Denmark	1
NLO Germany	1
NLO Italy	1
NLO UK	1
NLO US	1
NLR to SACLANT	
NLR Belgium	1
NLR Canada	1
NLR Denmark	1
NLR Germany	1
NLR Greece	1
NLR Italy	1
NLR Netherlands	1
NLR Norway	1
NLR Portugal	1
NLR Turkey	1
NLR UK	1
Total external distribution	250
SACLANTCEN Library	10
Stock	20
Total number of copies	280